Generating knowledge graphs by employing Natural Language Processing and Machine Learning techniques within the scholarly domain

Generating Knowledge Graphs by Employing Natural Language Processing and Machine Learning Techniques within the Scholarly Domain

Danilo Dessì[a,b,c,*], Francesco Osborne[d], Diego Reforgiato Recupero[a], Davide Buscaldi[e], Enrico Motta[d]

[a]*Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy*
[b]*FIZ Karlsruhe - Leibniz Institute for Information Infrastructure, Germany*
[c]*Karlsruhe Institute of Technology, Institute AIFB, Germany*
[d]*Knowledge Media Institute, The Open University, Milton Keynes, UK*
[e]*LIPN, CNRS (UMR 7030), University Paris 13, Villetaneuse, France*

Abstract

The continuous growth of scientific literature brings innovations and, at the same time, raises new challenges. One of them is related to the fact that its analysis has become difficult due to the high volume of published papers for which manual effort for annotations and management is required. Novel technological infrastructures are needed to help researchers, research policy makers, and companies to time-efficiently browse, analyse, and forecast scientific research. Knowledge graphs i.e., large networks of entities and relationships, have proved to be effective solution in this space. Scientific knowledge graphs focus on the scholarly domain and typically contain metadata describing research publications such as authors, venues, organizations, research topics, and citations. However, the current generation of knowledge graphs lacks of an explicit representation of the knowledge presented in the research papers. As such, in this paper, we present a new architecture that takes advantage of Natural Language Processing and Machine Learning methods for extracting entities and relationships from research publications and integrates them in a large-scale knowledge graph. Within this research work, we i) tackle the challenge of knowledge extraction by employing several state-of-the-art Natural Language Processing and Text Mining tools, ii) describe an approach for integrating entities and relationships generated by these tools, iii) show the advantage of such an hybrid system over alternative approaches, and vi) as a chosen use case, we generated a scientific knowledge graph including 109,105 triples, extracted from 26,827 abstracts of papers within the *Semantic Web* domain. As our approach is general and can be applied to any domain, we expect that it can facilitate the management, analysis, dissemination, and processing of scientific knowledge.

*Corresponding author. Tel. +39-070-675-8756.
Email address: danilo_dessi@unica.it (Danilo Dessì)

1. Introduction

Nowadays, we are seeing a constant growth of scholarly knowledge, making the access to scholarly contents more and more challenging through traditional search methods. This problem has been partially solved thanks to digital libraries which provide scientists with tools to explore research papers and to monitor research topics. Nevertheless, the dissemination of scientific information is mainly document-based and mining contents requires human manual intervention, thus limiting chances to spread knowledge and its automatic processing [1].

Despite the large number and variety of tools and services available today for exploring scholarly data, current support is still very limited in the context of sensemaking tasks that require a comprehensive and accurate representation of the entities within a domain and their semantic relationships. This raises the need of more flexible and fine-grained scholarly data representations that can be used within technological infrastructures for the production of insights and knowledge out of the data [2, 3, 4]. Kitano [5] proposed a similar and more ambitious vision, suggesting the development of an artificial intelligence system able to make major scientific discoveries in biomedical sciences and win a Nobel Prize.

Among the existing representations, knowledge graphs i.e., large networks of entities and relationships, usually expressed as RDF triples, relevant to a specific domain or an organization [6], provide a great method to organize information in a structured way. They already have been successfully used to understand complex processes in various domains such as social networks ego-nets [7] and biological functions [8].

Tasks like question answering, summarization, and decision support have already benefited from these structured representations. The generation of knowledge graphs from unstructured source of data is today key for data science and researchers across various disciplines (e.g., Natural Language Processing (NLP), Information Extraction, Machine Learning, and so on.) have been mobilized to design and implement methodologies to build them. State-of-the-art projects such as DBPedia [9], Google Knowledge Graph, BabelNet[1], and YAGO[2] build Knowledge Graphs by harvesting entities and relations from textual resources (e.g., Wikipedia pages). The creation of such knowledge graphs is a complex process that typically requires the extraction and integration of various information from structured and unstructured sources.

Scientific knowledge graphs focus on the scholarly domain and typically contain metadata describing research publications such as authors, venues, organizations, research topics, and citations. Some examples are Open Academic Graph[3], Scholarlydata.org [10], Microsoft Academic Graph[4] [11] (MAG), Scopus[5], Semantic Scholar[6],

[1] https://babelnet.org/

[2] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[3] https://www.openacademic.ai/oag/

[4] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph

[5] https://www.scopus.com/

[6] https://www.semanticscholar.org/

Aminer [12], Core [13], OpenCitations [14], and Dimensions[7]. These resources provide substantial benefits to researchers, companies, and policy makers by powering data-driven services for navigating, analyzing, and making sense of research dynamics. However, the current generation of knowledge graphs lacks of an explicit representation of research knowledge discussed in the scientific papers. This is usually only described by not machine-readable metadata, such as natural language text in the title and abstract, and in some cases a list of topics or keywords from a domain vocabulary or taxonomy (e.g., MeSH[8], ACM Digital Library[9], PhySH[10], CSO[11]). These data are useful to some degree, but do not offer a formal description of the nature and the relationships of relevant research entities. For instance this representation does not give us any information about what "sentiment analysis" is and how it interlinks with other entities in the research domain. It would be much more useful to know that this is a sub-task of Natural Language Processing that aims at detecting the polarity of users opinion by applying a range of machine learning approaches on reviews and social media data such as twitter posts.

A robust and formal representation of the content of scientific publications that types and interlinks research entities would enable many advanced tasks that are not supported by the current generation of systems. For instance, it would allow to formulate complex semantic queries about research knowledge such as "return all approaches and benchmarks that are used to detect fake news". It would also support tools for the exploration of research knowledge by allowing users to navigate the different semantic links and retrieve all publications associated with specific claims. It could also enable a new generation of academic recommendation systems and tools for hypothesis generation.

The Semantic Web community has been working for a while on the generation of machine-readable representations of research, by fostering the Semantic Publishing paradigm [15], creating bibliographic repositories in the Linked Data Cloud [16], generating knowledge bases of biological data [17], formalising research workflows [18], implementing systems for managing nano-publications [19, 20] and micropublications [21], and developing a variety of ontologies to describe scholarly data, e.g., SWRC [12], BIBO [13], BiDO[14], FABIO[15], SPAR[16], CSO[17], and SKGO[18] [23]. Some recent solutions, such as RASH[19] [24], and the Open Research Knowledge Graph[20] [25]

[7] https://www.dimensions.ai/
[8] https://www.nlm.nih.gov/mesh/meshhome.html
[9] https://dl.acm.org/
[10] https://physh.aps.org/
[11] https://cso.kmi.open.ac.uk/home
[12] SWRC - http://ontoware.org/swrc
[13] BIBO - http://bibliontology.com
[14] BiDO - http://purl.org/spar/bido
[15] FABIO - http://purl.org/spar/fabio
[16] SPAR - http://www.sparontologies.net/[22]
[17] CSO - https://cso.kmi.open.ac.uk/ [**?**]
[18] SKGO - https://github.com/saidfathalla/Science-knowledge-graph-ontologies
[19] RASH - https://github.com/essepuntato/rash
[20] ORKG - https://www.orkg.org/orkg/

highlighted the advantages of describing research papers in a structured manner. However, the resulting knowledge bases still need to be manually populated by domain experts, which is a time consuming and expensive process. We still lack systems able to extract knowledge from large collection of research publications and automatically generate a comprehensive representation of research concepts.

It follows that a significant open challenge in this domain regards the automatic generation of scientific knowledge graphs that contain an explicit representation of the knowledge presented in scientific publications [25], and describe entities such as approaches, claims, applications, data, results reported in each paper. The resulting knowledge base would be able to support a new generation of content-aware services for exploring the research environment at a much more granular level.

Most of the relevant information for populating such a knowledge graph might be derived from existing textual elements of research publications. To such an aim, in the last years, we assisted to the emergence of several excellent Machine Learning and NLP tools for entity linking and relationship extraction [26, 25, 27, 28, 29]. However, integrating the output of these tools in a coherent and comprehensive knowledge graph is still an open issue.

For instance, different tools may use different lexical resources, named-entity recognition approaches, and training sets and thus will often label the same entities with different names and disagree on the relation between them.

In this paper, we present a novel architecture that uses an ensemble of NLP and Machine Learning methods for extracting entities and relationships in form of triples from research publications, and then integrates them in a knowledge graph using Semantic Web best practices. The main hypothesis behind this work is that an hybrid framework combining both supervised and unsupervised methods will produce the most comprehensive set of triples (i.e., high recall) while still yielding a good precision.

Within our work, we refer to an entity as a statement that indicates an object (e.g., a topic, a tool name, a well-known algorithm, etc.). We create a relation between two entities when they are syntactically or semantically connected. As an example, if a tool T employs an algorithm A, we may build the triple $<T,\ employ,\ A>$. We compared our approach versus alternative methods on a manually annotated gold standard covering the Semantic Web domain.

The main contributions of the research presented in this paper are therefore the following:

- we propose an architecture that combines various tools for extracting entities and relations from research publications;
- we employ Semantic Web best practices, statistics, NLP, and Machine Learning techniques for integrating these entities and triples;
- we show the advantage of an hybrid approach versus methods that are only focused on supervised classification (e.g., Luan Yi et al. in [29]) or NLP tools (e.g., OpenIE);
- we carry out an evaluation of the resulting triples in terms of precision, recall, and F-measure;

- we generated a gold standard of manually annotated triples that can be used as benchmark for this task.

In this paper we focus on the Semantic Web as main domain, but the resulting approach is general and can be applied to any other domain. The code of the framework, the extracted triples, and the gold standard used in the evaluation are available through a GitHub repository[21].

The remainder of this paper is organized as follows. Section 2 formalizes the problem we addressed. The proposed methodology is detailed in Section 3. The evaluation and its discussion are reported in Section 4. Section 5 discusses the related work and highlights the main differences with the proposed approach. Finally, Section 6 concludes the paper, explains limitations that still exist, and defines future research works.

2. Problem Statement

Given a large collection of research papers, we want to generate a large-scale knowledge base, that will include all relevant entities in a certain domain and their relationships.

More in detail, given a set of scientific documents $D = \{d_1, \ldots, d_n\}$, we build a model $\gamma : D \rightarrow T$, where T is a set of triples (also referred as relationships) (s, p, o) where s and o belong to a set of entities E and p belongs to a set of relations labels L. Each triple needs also to be associated with the set of papers it was extracted from, allowing to assess how the claim is supported in the original collection of documents.

The resulting knowledge graph can be employed for different problems of new research fields (e.g., detection of research communities, their dynamics and trends, forecasting of research dynamics using sentiment analysis, measuring fairness of open access datasets, etc.), and, in general, as a support resource for scientists in conducting scientific research.

3. Methodology

In this section, we describe the approach that we applied to produce a scientific knowledge graph of research entities. The workflow of our pipeline is shown in Figure 1. In short, our framework includes the following steps:

1. **Extraction of entities and triples**, which exploits an ensemble of several NLP and machine learning tools to extract triples from text.
2. **Entity refining**, in which the resulting entities are merged and cleaned up.
3. **Triple refining**, in which the triples extracted by the different tools are merged together and the relations are mapped to a common vocabulary.
4. **Triple selection**, in which we select the set of ”trusted” triples that will be included in the output by first creating a smaller knowledge graph composed by triples associated with a good number of papers and then enriching this set with

[21] `https://github.com/danilo-dessi/skg`

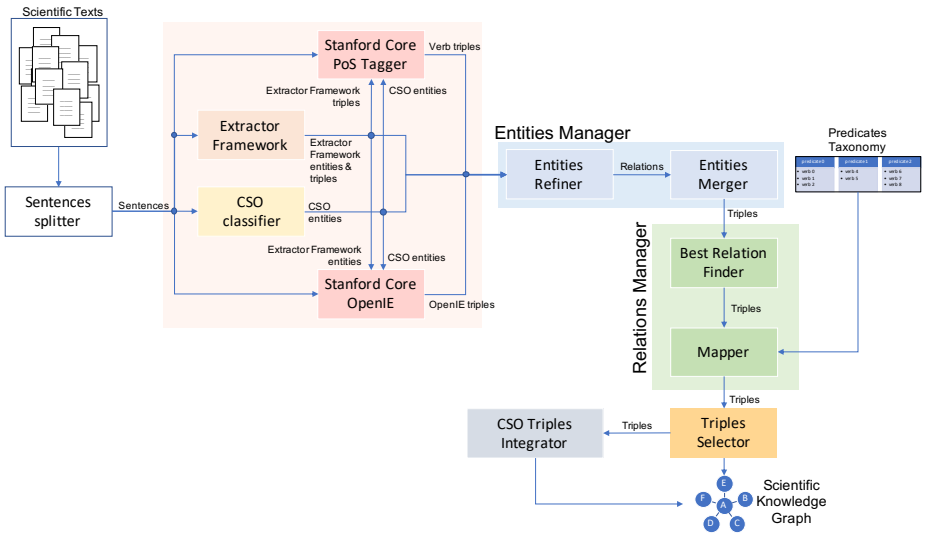

Figure 1: Workflow of our approach for building a scientific knowledge graph from scientific textual resources.

other semantically consistent triples. In the following subsection we will describe the architecture in more details and discuss the specific NLP and machine learning tools that we used in the implementation of our prototype.

3.1. Extraction of Entities and Relations

For extracting entities and relations, we exploited the following methods:

- *The extractor framework [29]* designed by Luan Yi et al. that we modified and embedded within our pipeline. It is based on Deep Learning models and provides modules for detecting entities and relations from scientific literature. It detects six types of entities (*Task, Method, Metric, Material, Other-Scientific-Term, and Generic*) and seven types of relations among a list of predefined choices (*Compare, Part-of, Conjunction, Evaluate-for, Feature-of, Used-for, Hyponym-Of*). For the purpose of this work, we discarded all the triples with relation *Conjunction*, since they were too generic. In particular, the extractor framework uses feed-forward neural networks over span representations of the input texts to compute two scores v_1 and v_2. The score v_1 is computed on single spans and measures how likely a span may be associated to an entity type. The second score v_2 is a pairwise score on a pair of span representations and measures how likely spans are involved in a relation. Therefore, for a given pair of span representations, let's say (t_1, t_2), the scores $v_1^{t_1}$, $v_1^{t_2}$, and $v_2^{(t_1,t_2)}$ are computed. If both $v_1^{t_1}$ and $v_1^{t_2}$ meet a threshold t_{entity}, and $v_2^{(t_1,t_2)}$ meets a threshold $t_{relation}$ then the span representations t_1 and t_2 are labelled as entities, and their pair as relationship (t_1, t_2). The type of entity where the value v_1 is the highest is associated to the entity

itself. Similarly, a pair (t_1, t_2) is associated to the type of relation r where the pair has the highest value of v_2, yielding the triple (t_1, r, t_2). For example, from the following sentence "*We propose a new web recommendation system based on reinforcement learning.*", this framework detected *web recommendation system* as a *Task*, *reinforcement learning* as a *Method*, and the relation *Used-for* between them, yielding the triple *<reinforcement learning, Used-for, web recommendation system >*. We refer to this framework as Extractor Framework.

- *The CSO Classifier [30]*[22], a tool for automatically classifying research papers according to the Computer Science Ontology (CSO)[23] [31], which is a comprehensive automatically generated ontology of research areas in the field of Computer Science. The current version of CSO describes 14K research topics arranged in a nine level polyhierarchical taxonomy. The CSO classifier identifies topics by means of two different components, the syntactic module and the semantic module. The *syntactic module* removes English stop words and collects unigrams, bigrams, and trigrams. Then, for each n-gram, it computes the Levenshtein similarity with the labels of the topics in CSO. Finally, it returns all research topics whose labels have a similarity score equal to or higher than a threshold to one of the n-grams. The *semantic module* uses part-of-speech tagging to identify candidate terms composed of a proper combination of nouns and adjectives and maps them to the ontology topics by using a Word2Vec model trained on titles and abstracts of 4,5M English papers in the field of Computer Science from MAG. Then, the module computes a relevance score for each topic in the ontology by considering the number of times the topic was identified within the retrieved words. The CSO classifier combines the outputs of these two modules and enhances the resulting set by including all relevant super-topics according to the *superTopicOf*[24] relationship in CSO. For instance, if an article was tagged with the topic *neural networks*, it would also be associate to its super-topics *machine learning*, *artificial intelligence*, and *computer science*. This latter functionality is not used in the extraction stage, since it would not be possible to map super-topics with other entities in the sentence. However, we used a similar process in the last phase of the process (Knowledge Graph Enhancement, see Section 3.5) for generating further triples by exploiting CSO hierarchical relationships.

- *OpenIE [32]* provided by the Stanford Core NLP suite. It detects general entities and relations among them. Relations are detected by analyzing clauses (i.e., groups of words that contain at least a subject and a verb) which are built by exploring the parse tree of the input text. In the first stage, the methodology produces clauses from long sentences which stand on their own syntactically and semantically. For doing so, it uses a multinomial logistic regression classifier to recursively explore the dependency tree of sentences from governor to dependant

[22] `https://github.com/angelosalatino/cso-classifier`
[23] `http://cso.kmi.open.ac.uk`
[24] `https://cso.kmi.open.ac.uk/schema/cso`

nodes. Then, it applies logical inferences to capture natural logic within clauses by using semantics dictating contexts. Doing so, OpenIE is able to replace lexical items with something more generic or more specific. Once a set of short entailed clauses is produced, it segments them into its output triples. In our approach we keep triples where entities match those found by the Extractor Framework and the CSO Classifier, so that we caught only those triples that refer to entities of the target domain.

- *The Stanford Core NLP PoS tagger*[25] which extracts predicates between the entities identified by the Extractor Framework and the CSO Classifier. More specifically, for each sentence s_i it detects all verbs $V = \{v_0, \ldots, v_k\}$ between each pair of entities (e_m, e_n) of that sentence and generates triples in the form $<e_m, v, e_n>$ where $v \in V$.

  Our goal was to detect the most used verbs between two entities at the cost of producing some noisy relations. Indeed, this approach is able to return several additional relationships that were missed by the other tools. In the following sections we describe how we handled and validate these triples in order to reduce the noise.

We processed each sentence from all the abstracts and used the tools and methods above to assign to each sentence s_i a list of entities E_i and a list of triples R_i.

First, we run the extractor framework to extract both entities E_i and triples R_i. Secondly, we used the CSO Classifier to extract all Computer Science topics, further expanding E_i. Thirdly, we processed each sentence s_i with OpenIE, and retrieved all the triples composed by subject, verb, and object in which both subject and object matched the entities resulting from the previous steps. Finally, for each sentence s_i we took all the verbs within two entities through the PoS Tagger, yielding R_i thoroughly expanded.

3.2. Entities Manager

During the extraction process it might happen that different entities in E_i may actually refer to the same concept with alternative forms, or may represent too generic concepts that do not carry meaningful information. In this section, we briefly describe which steps we have performed by the Entities Refiner and Entities Mapper modules in order to address these issues.

3.2.1. Entities Refiner Module

Many of the entities resulting from previous steps can be noisy, ambiguous, and too generic.

For example, entities like “approach” and “method” are too abstract and thus not very useful for our purpose. Their presence simply add noise to the the knowledge graph.

[25] `https://nlp.stanford.edu/software/tagger.shtml`

The goal of this module is to preprocess the entities, merging alternative labels, discarding ambiguous and generic entities, and splitting the ones that include compound expressions.

Cleaning up entities. First, we removed punctuation (e.g., dots and apostrophes) and stop-words (e.g., pronouns) from entities. We also removed some words that might be mixed up (e.g., *it* might be the pronoun *it* or the acronym of *information technology*) by using a blacklist.

Splitting entities. Some entities actually contained multiple compound expressions, e.g., *Machine Learning and Data Mining*. Therefore, we split entities that contain the conjunction *and*. Referring to our example, we obtained the two entities *Machine Learning* and *Data Mining*.

Handling Acronyms. Acronyms are usually defined, appearing the first time near their extended form (e.g., *Web Ontology Language (OWL)*) and then by themselves in the rest of the abstract (e.g., *CSO*). In order to map acronyms with their extended form in a specific abstract we use a regular expression. We then substituted every acronym (e.g., *OWL*) in the abstract with their extended form (e.g., *Web Ontology Language*). Since acronyms can be ambiguous, we perform this operation only on entities from the same abstract.

Detection of Generic Entities. Entities might be too generic for the purpose to describe the knowledge of a domain (e.g., *content*, *time*, *study*, *article*, *input*, and so on). We discard these kind of entities by applying a frequency-based filter which compares their frequency in three sets of documents:

- the set of publications of the Semantic Web.
- a set of the same size covering *Computer Science* domain, but not *Semantic Web*.
- a set of the same size containing papers from various domains, but not about *Semantic Web* nor the *Computer Science*.

For each entity e, we computed the number of times it appeared in the above datasets, so that we had three different counts $c_e^{'}, c_e^{''}, c_e^{'''}$. We normalized the counts by dividing them with the number of words of the set where they were computed. Then we computed the ratios $r_e^{'} = \frac{c_e^{'}}{c_e^{'''}}$ and $r_e^{''} = \frac{c_e^{'}}{c_e^{''}}$. If the ratio $r_e^{'}$ met a threshold $t_e^{'} = 2$, and the ratio $r_e^{''}$ met a threshold $t_e^{''} = 10$, the entity e was included in the graph. Thresholds were empirically defined by manually evaluating which entities were saved/discarded.

In addition, we automatically preserved all the entities within a whitelist that includes CSO topics and the author's keywords of all the papers in the input dataset.

3.2.2. Entities Merger Module

We merge entities with the same meaning by using both a lemmatizer and the CSO ontology. Singular and plural forms are combined by using the Lemmatizer available in the SpaCy[26] library. Then we exploited the alternative labels described by CSO to merge entities that refer to the same research topic (e.g., "ontology alignement" and

[26] `https://spacy.io`

"ontology matching"). More specifically, given an entity $e \in E$ that is known by CSO, let $A_e = \{e_0, ..., e_{k-1}\}$ be the set of alternatives of e in CSO. The module first finds the longest label $e_{longest} \in A_e$, then e is replaced by $e_{longest}$. The same process is repeated for each entity $f \in E$.

3.3. Relations Manager

This step aims at (i) finding the best relation predicate for each pair of entities e_i, e_j where a relation exists (each element in R), and (ii) mapping all the relations within a table we have defined.

3.3.1. Best Relation Finder Module

Here, the set of triples R presents three different types of triples: those extracted by the Extractor Framework, let us say R_{EF}, those coming from OpenIE, let us say R_{OIE}, and those detected with the PoS tagger, called R_{PoS}. We performed the following operations on these sets:

- On the set of triples in R_{EF} we acted as follows. Given a pair of entities (e_p, e_q) in R_{EF}, we merged into a list L_r all relations' labels r_i such that $(e_p, r_i, e_q) \in R_{EF}$. Then we chose the most frequent relation $r_{most\_frequent} \in L_r$, and built a single triple $(e_p, r_{most\_frequent}, e_q)$. Triples so built formed the set T_{EF}. Clearly, the size of the set T_{EF} is lower than the size of the set R_{EF}.

- On the set R_{OIE} we performed a deeper merging operation. Similarly to the work performed on R_{EF}, given a pair of entities (e_p, e_q) in R_{OIE}, we first merged into a list L_r all relations' labels r_i such that $(e_p, r_i, e_q) \in R_{OIE}$. In R_{OIE} all triples have a verb as relation predicate. Hence, we assigned each r_i to its word embedding w_i from the word embeddings built on the MAG dataset, yielding the list L_w. With the word embeddings in L_w an averaged word embedding w_{avg} was built. Then, the relation r_i with the word embedding w_i nearest to w_{avg} according to the cosine similarity was chosen as final relation for the pair (e_p, e_q), yielding the triple (e_p, w_i, e_q). The same procedure was also applied on R_{PoS}. The execution of this procedure on R_{OIE} and R_{PoS} yielded the sets T_{OIE} and T_{PoS}, respectively.

- Finally, for the sets T_{EF}, T_{OIE}, and T_{PoS}, we saved for each triple (e_p, r_i, e_q) the number of papers where the pair of entities (e_p, e_q) appeared. We refer to this number as the *support* of triples.

3.3.2. Mapper Module

From the previous step, a large number of verb relations resulted. However, the majority of relations have a common meaning with others, i.e., many relations were represented by synonyms. For example, the relations *uses*, *utilizes*, *adopts*, and *employs* may be used to express the same concept within a triple with only a slight change in meaning. Within our triples set, there were a good number of triples that represented the same information such as <*ontology alignment, uses, ontology*> and <*ontology alignment, utilizes, ontology*>. Hence, in order to reduce the number of redundant relations, we built a map $M : verb\_relation \rightarrow verb\_relation_{representative}$ where semantically similar relations were mapped to a single label (e.g., the relations *uses*, *utilizes*,

adopts, *employs* were mapped to the same representative relation *uses*). In order to do this, we first retrieved all word embeddings that represented verb relations from sets T_{OIE} and T_{PoS}. The rationale behind this is that word embeddings represent semantic and syntactic properties of the words and, therefore, verb relations with similar word embeddings have similar semantics and meaning. Then, we used the hierarchical clustering algorithm provided by the SciKit-learn library[27], which uses (1− cosine similarity) as distance to group together similar verb relations. The cosine similarity quantifies the angle between two vectors. Its formula applied on two vectors v_1 and v_2 can be observed in (1).

$$Cosine\_similarity(v_1, v_2) = \frac{v_1 \cdot v_2}{\| v_1 \| \cdot \| v_2 \|} \tag{1}$$

The resulting clustering dendrogram was cut by an empirically determined threshold of the averaged Silhoutte-width = 0.65. Values of Silhouette width range from −1 to 1. When the value is closer to 1, it means that the clusters are well separated; when the value is closer to 0, it might be difficult to detect the decision boundary; when the value is closer to -1, it means that elements assigned to a cluster might have been erroneously assigned. Its formula is computed as shown in (2), where given a cluster c, $w(c)$ represents the average dissimilarity of elements in c, and $o(c)$ is the lowest average dissimilarity of elements of c to any other cluster.

$$s(c) = \frac{o(c) - w(c)}{max\{o(c), w(c)\}} \tag{2}$$

Subsequently, we manually revised the clusters and built the map M, where each verb relations of each cluster was mapped to the representative relation identified by the cluster centroid. For example, each verb relation of the cluster {*builds*, *creates*, *produces*, *develops*, *makes*, *constructs*, etc.} was mapped to the centroid *produces*. Finally, the relations used in the set T_{EF} were manually integrated within the map M. All triples from the union of T_{EF}, T_{OIE}, and T_{PoS} were mapped by using M (e.g., the triple <*knowledge construction, creates, ontology integration platform*> was transformed in <*knowledge construction, produces, ontology integration platform*>), so that a well-defined set of relations was used within our final resulting triples.

3.4. Triples Selection

In this section the method we employed to choose only certain triples is presented. We also define what we mean with the words *valid* and *consistent* associated to our triples in order to build the scientific knowledge graph.

3.4.1. Valid Triples

For the purpose of including meaningful triples within our knowledge graph, we first define a smaller knowledge graph composed of ”valid” triples. These can be defined in different way according to the performance of the tools in the first step and the number of papers supporting a certain triples.

In the current prototype we define as valid the following triples:

[27] `https://scikit-learn.org`

- We consider *valid* all the triples obtained by the Extractor Framework (T_{EF}) and the OpenIE tool (T_{OIE}).
- All triples associated with at least 10 papers (indicating a fair consensus). Therefore, we consider *valid* the triples that were detected by the PoS tagger associated with at least 10 papers. We refer to this set as T'_{PoS} such that $T'_{PoS} \subseteq T_{PoS}$.

The union of T_{TF}, T_{OIE}, and T'_{PoS} composed the set of all valid triples T_{valid}.

3.4.2. Consistent Triples

The set of triples not in T_{valid}, that we label $T_{invalid}$, may still include several good triples that were not associated to sizable number of papers. More specifically, consensus of the community about scientific claims is built over time and, hence, new discoveries might not have a high support. However, these triples are still important since they can suggest the ways to go along to rapidly explore the most recent research trends. We thus use the triples in T_{valid} as examples to learn which triples are consistent with the valid ones and could still be included in the final outcome. Specifically, we trained a classifier $\gamma : P \rightarrow L$ where P is a set of pair of entities blue in T_{valid} and L is the set of relations used in M (e.g., *uses*, *provides*, *supports*, *improves*), with the aim of comparing the actual relation with the one returned by the classifier. The intuition is that a triple consistent with T_{valid} would have its relation correctly guessed by the classifier. In order to do so, we performed the following steps:

1. We generated word embeddings of size 300 by processing with the Word2vec algorithm [33, 34] all the input abstracts. For multi-word entities we replaced white spaces with underscore characters within our abstracts texts (e.g., the entity *semantic web* becomes *semantic_web*).
2. We trained a Multi-Perceptron Classifier (MLP) to return the relation between a couple of entities. We used the concatenation of the embeddings of subject and object entities as input and the relation as output.
3. The validation step was performed by applying the classifier on all the triples (e_p, r, e_q) in $T_{invalid}$ and comparing the actual relation r with the relation returned by the classifier r'. If $r = r'$ then the triple (e_p, r, e_q) was considered valid and added to T_{valid}. Otherwise we computed the cosine similarity cos_sim and the *Wu-Palmer*[28] similarity between the embedding of r and r'. If the average between cos_sim and wup_sim was higher than a threshold t (empirically set at 0.5) then the triple (e_p, r, e_q) was considered valid and added to T_{valid}.

3.5. Knowledge Graph Enhancement

In order to increase the amount of the resulting information, we added to the produced knowledge graph the additional triples that could be inferred by exploiting the hierarchical relations in CSO. More precisely, given a triple (e_2, r, e_1), if in CSO the entity e_3 is *superTopicOf* of the entity e_1 and there is no triple involving e_2 and e_3, we

[28] `http://www.nltk.org/howto/wordnet.html`

Table 1: Examples of triples that our pipeline detects. In *Italic* some examples of triples that were discarded by our pipeline.

Subject Entity	Relation	Object Entity
semantic web technologies	supports	contextual information
semantic relationship	defines	ontologies
structural index	uses	structural graph information
thesaurus	hyponymy-of/is	knowledge organization system
web page classification	uses	text of web page
question answering systems	uses	semantic relation interpreter
context models	*proposes*	*web ontology language*
data exchange	*queries*	*web ontology language*
domain-specific ontologies	*executes*	*semantic search engines*
fuzzy logics	*maintains*	*semantic descriptions*
learning objects	*learns*	*semantic web services*
resource description framework (rdf)	*uses*	*digital libraries*

also infer the triple (e_2, r, e_3). For instance, given the triple <*nlp systems, uses, named-entity recognition* >, if *artificial intelligence* is *superTopicOf named-entity recognition*, we can infer the triple <*nlp systems, uses, artificial intelligence* >.

This last step was performed by the CSO Triples Integrator module in the pipeline. Finally, the triples are converted to RDF and returned.

4. Results and Discussion

This section details the scientific knowledge graph we have produced and shows how we have validated it.

4.1. The Semantic Knowledge Graph

Here we report the result of our framework, focusing on the *Semantic Web* domain.

We used an input dataset composed by 26,827 abstracts of scientific publications about this domain that was retrieved by selecting publications from the Microsoft Academic Graph dataset[29]. It is a knowledge graph related to the scholarly domain that describes more than 200 million scientific publications through metadata such as title, abstract texts, authors, venue, field of study and so on. For our purpose we considered only abstracts that were classified under Semantic Web by the CSO Classifier [30]. This dataset has also been used for exploring the relationship between Academia and Industry by Angioni et al. [35].

A few examples of retrieved triples as well as of triples that were discarded by our pipeline can be seen in Table 1.

[29] `https://www.microsoft.com/en-us/research/project/microsoft-academic-graph`

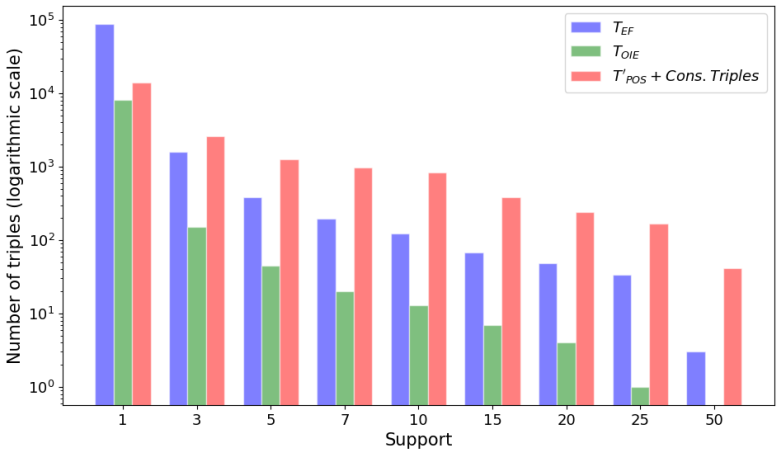

Figure 2: Comparison of the distribution of the support of the three methods.

The resulting knowledge graph includes 109,105 triples: 87,030 from the Extractor Framework (T_{EF}), 8,060 from OpenIE (T_{OIE}), and 14,015 from the PoS tagger method and classifier (T'_{PoS} + *Cons. Triples*).

However, the raw number of triples extracted by each method can be misleading. In fact, some triples are supported by a large number of papers, suggesting a large consensus of the scientific community and more in general a claim that can easily be trusted, while some other appear in one or very few papers

Figure 2 reports the distribution of the support of the triples produced by T_{EF}, T_{OIE} and T'_{PoS} + *Cons. Triples*.

While T_{EF} produces the most sizable part of those triples, most of them have a very low support. In fact, 80,030 of them are supported by a single paper and 1,580 by only three papers. They may thus contain claims that did not reach yet a consensus in the community. For all the other support values, the set T'_{PoS} + *Cons. Triples* has a higher number of triples than T_{EF} and T_{OIE} and, hence, it is possible to assume that T'_{PoS} triples may be more in accordance within the community of *Semantic Web*. For instance, if we take in consideration only the triples whose support is equal or greater than 5, only 393 triples are provided by the set T_{EF}, 45 by T_{OIE} and, 1,268 by T'_{PoS} + *Cons. Triples*. It is also worth to note that when the support is very high (e.g., equal or greater than 50) there are not triples provided by the set T_{OIE}, and few triples provided by T_{EF}. This still stresses the fact that those triples might not express valuable knowledge or have consensus within the *Semantic Web* community.

4.2. Gold Standard Creation

We first used several different approaches to generate triples from the 26,827 abstracts described in the previous section. Specifically, we applied on this dataset: 1)

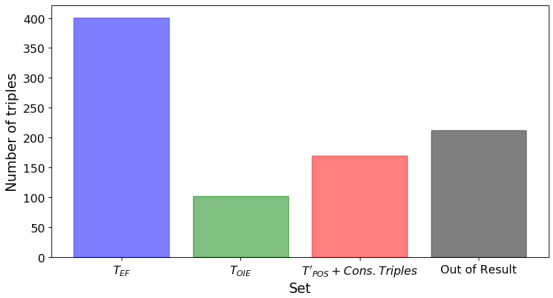

Figure 3: Distribution of triples within the gold standard.

T_{EF} (i.e., the Extractor Framework), 2) T_{OIE} (i.e., OpenIE), 3) T'_{PoS} (considering only the triples with support ≥ 10), and T'_{PoS} + *Cons. Triples*.

The resulting set of 109, 105 triples would be unfeasible to manually annotate, since it is very large and includes terms related to very different areas of expertise. We thus focused only on 818 triples which contain (as subject or object) at least one of the 24 sub-topics[30] of Semantic Web and at least another topic in the CSO ontology. This set contains 401 triples from T_{EF}, 102 from T_{OIE}, 60 triples from T'_{PoS} and 110 relevant *Cons. Triples*. In order to measure the recall, we also added 212 triples that were discarded by the framework pipeline. The reader notices that the total number of triples (818) is slightly less than the sum of various sets (401+102+60+110+212) because some triples have been derived by more than one tool. The triples distribution of the gold standard can be observed in Figure 3.

We recruited five researchers in the field of Semantic Web and asked them to annotate each triple either as *true* or *false*. In order to do so, they assessed each triple according to their expertise of the field. They were also allowed to search concepts on the web and in the literature when they were not familiar with a specific entity. The averaged agreement between experts was 0.747 ± 0.036, which indicates a high inter-rater agreement. We then created the gold standard using the majority rule approach. Specifically, if a triple was considered relevant by at least three annotators, it was labeled as true, otherwise as false.

The purpose of this gold standard is twofold. First, it allows us to evaluate the proposed pipeline to extract triples from scholarly data and, second, it provides a resource which will facilitate further evaluations.

4.3. Precision, Recall, F-measure Analysis

For evaluating our methodology, we performed a precision, recall, F-measure analysis considering various combinations of relations sources. Measures are computed as shown by equations (3), (4), and (5).

[30]There exist 24 sub-topics of Semantic Web within the CSO ontology.

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP + FN} \tag{4}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{5}$$

In equations (3) and (4) TP (true positive) indicates the number of triples labelled as *good* and returned by our pipeline, FN (false negative) is the number of triples that were labelled as *good* but not returned by our pipeline, and FP (false positive) is the number of triples that have been erroneously returned by our pipeline (i.e., triples were labelled as *bad* in the gold standard but our pipeline picked up them as *good* triples). The F-measure is computed as the harmonic mean of (3) and (4) as shown in equation (5).

We tested eight alternative approaches:

- The Extractor Framework from Luan Yi et al. [29] (**EF**) described in section 3.1.
- OpenIE, from Angeli et al. [32] (**OpenIE**) described in section 3.1.
- the Stanford Core NLP PoS tagger described in section 3.1, after merging the relevant triples as described in section 3.3.1 (T'_{PoS}). We considered only the triples with support ≥ 10.
- The previous approach enriched by consistent triples as described in section 3.4.2 (T'_{PoS} **+ Cons. Triples**).
- The combination of EF and OpenIE (**EF + OpenIE**).
- The combination of EF and T'_{PoS} + Cons. Triples (**EF +** T'_{PoS} **+ Cons. Triples**).
- The combination of OpenIE and T'_{PoS} + Cons. Triples (**OpenIE +** T'_{PoS} **+ Cons. Triples**).
- The final framework that integrates all the previous methods (**OpenIE + EF +** T'_{PoS} **+ Cons. Triples**).

Table 2 reports precision, recall, and F-measure of all the methods.

EF obtains an high level of precision (84.3%), but a recall of only 54.4%. OpenIE and T'_{PoS} shows a slightly lower level of precision and an even lower recall. T'_{PoS} + Cons. Triples obtains the best precision of all the methods (84.7%), highlighting the advantages of using a classifier for selecting consistent triples. Overall, all these basic methods produce triples with good precision, but suffer in term of recall.

Combining them together generally raises the recall without paying too much in term of precision. EF + OpenIE yields a F-measure of 72.8% with a recall of 65.1% and EF + T'_{PoS} + Cons. Triples a F-measure of 77.1% with a recall of 71.6%. The final version of our framework, which combines all the previous methods, obtains the best recall (80.2%) and F-measure (81.2%) and yields also a fairly good precision (78.7%).

This seems to confirm the hypothesis that an hybrid framework combining supervised and unsupervised methods would produce the most comprehensive set of triples and the best performance overall.

4.4. Examples and considerations about the Scientific Knowledge Graph

In this section, we show some sample of the triples extracted for the Semantic Web Knowledge Graph and discuss benefits and limitations of our output.

Table 3 shows a selection of the triples about the research topic *ontology alignment*, ranked by *support*. It is easy to see that many of these triples define the fundamental characteristics of *ontology alignment*. The topic is contextualized (via "skos:broader" relations) within the areas of *semantic web technologies* and *information integration*. *Ontology alignment* is defined as an entity that uses *ontologies*, selects *semantic correspondences*, and supports *semantic interoperability*.

Several other triples add further details, such as that *ontology alignment* finds *semantically related entities*, adopts *semantic similarity measures*, and limits the need for *human intervention*. Naturally, the representation also suffers from some issues that we plan to address in future work. For instance, the triples <*ontology alignment, selects, mapping*> and <*ontology alignment, supports, semantic relations*> appear too ambiguous. This may be either a limitation of our vocabulary of relations or an issue in the methodology used for merging together the triples from the PoS tagger. Similarly, in <*ontology alignment, produces, semantic web application*> the predicate does not appear to be correct, maybe "support" would be a better choice in this case. We thus plan to work further on our approach for merging triples and select the best predicate between two entities.

The triple <*ontology alignment, produces, semantic web application* > shows another typical issue. In the knowledge graph we have both "distributed and heterogeneous ontology" and "heterogeneous ontology" but no link between the two. In the future we need to be able to detect that "distributed and heterogeneous ontology" is actually a sub-concept of "heterogeneous ontology".

Table 2: Precision, Recall, and F-measure of each method adopted to extract triples. To note that the last row identified the triples extracted using the full pipeline.

Triples identified by	**Precision**	**Recall**	**F-measure**
EF	0.8429	0.5443	0.6615
OpenIE	0.7843	0.1288	0.2213
T'_{PoS}	0.8000	0.0773	0.1410
T'_{PoS} + Cons. Triples	**0.8471**	0.2319	0.3641
EF + OpenIE	0.8279	0.6506	0.7286
EF + T'_{PoS} + Cons. Triples	0.8349	0.7166	0.7712
OpenIE + T'_{PoS} + Cons. Triples	0.8145	0.3253	0.4649
OpenIE + EF + T'_{PoS} + Cons. Triples	0.7871	**0.8019**	**0.8117**

Table 3: Examples of triples from the Semantic Web Knowledge Graph.

Subject Entity	Relation	Object Entity	Support
ontology alignment	uses	ontologies	194
ontology alignment	skos:broader	semantic web technologies	65
ontology alignment	selects	semantic correspondence	45
ontology alignment	supports	semantic interoperability	34
ontology alignment	maintains	heterogeneous ontology	25
ontology alignment	selects	mapping	21
ontology alignment	selects	semantically related entity	19
ontology alignment	supports	semantic relation	17
ontology alignment	produces	semantic web application	14
ontology alignment	combines	concept similarity	13
ontology alignment	supports	semantic heterogeneity problem	13
ontology alignment	limits	human intervention	12
ontology alignment	executes	semantic similarity measures	12
ontology alignment	produces	ontology mapping method	11
ontology alignment	provides	distributed and heterogeneous ontology	10
ontology alignment	skos:broader	information integration	10
ontology alignment	uses	mapping system	10
ontology alignment	provides	matching technique	10

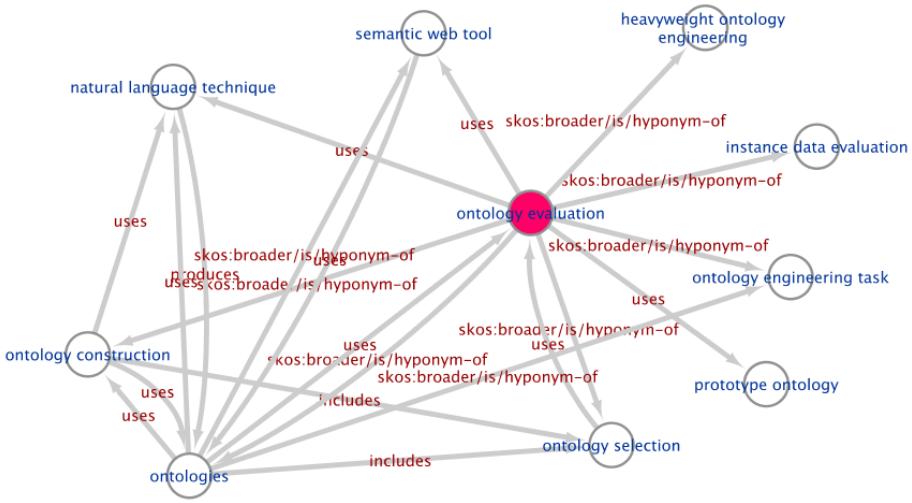

Figure 4: The subgraph of the entity "ontology evaluation" with related relationships in our Scientific Knowledge Graph within the Semantic Web domain.

Figure 4 shows a graphical representation of the research topic *ontology evaluation*. It is interesting to notice how this representation is also fairly interpretable by human users. Some examples about the information that can be derived includes:

- *ontology evaluation* uses *natural language techniques*. It suggests that there might be tools or methodologies that exploit textual resources written in natural language that have been involved in ontologies evaluation.
- *ontology evaluation* is hyponym of the entity *ontology construction* indicating that a specialized task within *ontology construction* involves the evaluation of the produced ontologies.
- *ontology evaluation* is hyponym of *instance data evaluation* which shows in which more general task the *ontology evaluation* falls.

Finally, it is also interesting to consider an entity that is not so much represented in the input dataset. Figure 5 shows the subgraph of the entity *supervised machine learning*. This representation is useful to highlight which topics and kind of resources are employed by *supervised machine learning* within the *Semantic Web* domain. As an example, it is easy to see that this entity uses both *structured data model* and *rich semantics*, and how these two entities are related as well. In the example, only two types of relations appear (i.e., *uses* and *includes*). They seem too generic, in fact, it is not clear how *supervised machine learning* adopts the other linked entities. This can indicate that our taxonomy of predicates may be too general and we may have to adopt a more fine grained representation in future work.

Overall, the knowledge graph seems to contain triples of good quality that well represent the main characteristics of research entities within the context of the input

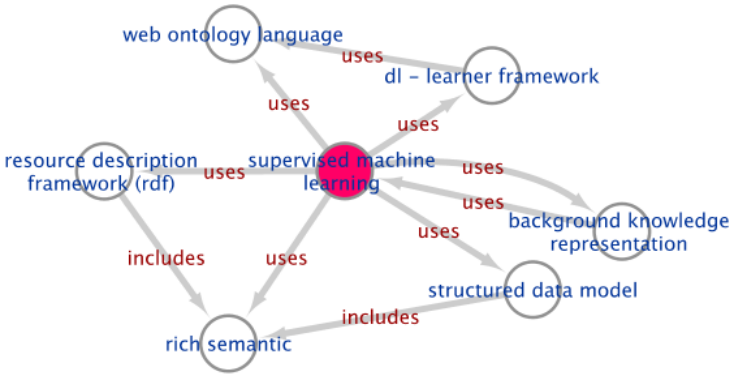

Figure 5: The subgraph of the entity "supervised machine learning" with related relationships in the produced Scientific Knowledge Graph within the Semantic Web domain.

dataset. We thus believe that this version may already be used for enhancing the representation of research items and supporting users in understanding and navigating research outcomes.

Specifically, we see four main applications of the knowledge graph. The first regard intelligent systems for navigating research publications, such as Open Knowledge Maps[31], which could further characterize entities according to their types and relationships and thus interlinking articles according to a variety of new facets and generating more semantically consistent clusters of articles. The knowledge graph should also be of interest for the growing area of graph embeddings. Indeed, we received several queries by research groups interested in running methods for producing graph embeddings on our output in order to generate a representation of the research entities that could be easily fed to machine learning algorithms and link detection techniques. Systems for recommending research papers (e.g., Mendeley[32], CORE[33]) could also take advantage of this knowledge base for improving and explaining their suggestions according to the entities in the articles. Finally, trend detection systems (e.g., Augur [36], ResearchFlow [37]), which typically identify entities of interest from a vocabulary or a domain taxonomy and monitor them across time, will benefit by having a large knowledge base of well-defined and interlinked research entities.

4.5. Applicability in other domains

In this section, we discuss which the current limitations of our pipeline to be used within other domains are, and suggest some developments that are required to capture other domain peculiarities. To start with, the current version of the proposed pipeline exploits both computer science-tuned modules (e.g., the extractor framework and the CSO classifier) and others more general ones that do not depend on the domain (e.g.,

[31] https://openknowledgemaps.org/
[32] https://www.mendeley.com/
[33] https://core.ac.uk/

OpenIE and the PoS Tagger extractors). Moreover, the handling of entities and relations does not depend on the target domain and, therefore, the pipeline is limited to the computer science field only by some tools employed in the extraction phase. More specifically, the extractor framework was trained on a corpus of computer science scientific papers, and the ontology employed only embraces computer science topics. As a matter of principle, this implies that the current pipeline can be exploited in any computer science sub-field without limitations. To use the pipeline on other domains, the main challenges are the substitution of the Extractor Framework and the Computer Science Ontology. However, this limitation can be easily tackled in many domains where there already exist tools that can be used to parse the domain scientific resources. To name an example, the *SciSpacy*[34] model can be used to parse scientific text within the biomedical domain to detect the research entities that characterize it. In the same way, most scientific disciplines offer domain ontologies or taxonomies that could be used in alternative to CSO. These includes the Medical Subject Heading (MeSH)[35] and *SNOMED-CT*[36] in Biology, the Mathematics Subject Classification (MSC)[37] in Mathematics, and the Physics Subject Headings (PhySH)[38] in Physics. Broadly speaking, the tools we used to detect the first entities and relations can be replaced by existing tools that have been already developed in other domains to capture domain specific information. One more point to be considered is that ontological resources are today being developed for many specific domains and use cases such as the Cultural Heritage domain (e.g, ArCO [38]), Robotics [39], Bio-Medicine[39], Computer Science (e.g., AIDA [40]), and so on. Therefore, the main efforts might be due to the developments of interfaces to feed our pipeline with new extraction resources output.

5. Related Work

Many information extraction approaches for harvesting entities and relationships from textual resources can be found in literature.

First, entities in textual resources have been detected by applying Part-Of-Speech (PoS) tags. An example is constituted by [41], where authors provided a graph based approach for Word Sense Disambiguation (WSD) and Entity Linking (EL) named Babelfly. Later, other approaches started to exploit various resources (e.g., context information and existing knowledge graphs) for developing ensemble methodologies [28]. Following this idea, we exploited an ensemble of tools to mine scientific publications and get information out of them. Then, we designed and implemented a software pipeline for the purpose of creating a scientific knowledge graph that organizes entities and their relations. Relations extraction is not a novel task and has been already addressed in literature in order to connect information coming from different pieces of

[34] `https://allenai.github.io/scispacy/`
[35] Medical Subject Heading - `https://www.ncbi.nlm.nih.gov/mesh`
[36] `http://www.snomed.org/snomed-ct/five-step-briefing`
[37] Mathematics Subject Classification - `https://mathscinet.ams.org/msc`
[38] Physics Subject Headings - `https://physh.aps.org/`
[39] `https://bioportal.bioontology.org/ontologies`

text. FRED[40] is a machine reader developed by [27] on top of Boxer [42]. It links elements to various ontologies in order to represent the content of a text in a RDF representation. Among its features FRED extracts relations between frames, events, concepts and entities. However, integrating its extracted knowledge for specific domain applications still remains an open challenge due to the unpredictable and too generic type of knowledge that is extracted, making difficult the use of its entities and relations for modelling scholarly contents. Moreover, FRED only considers a single text at a time and does not consider domain dependent characteristics that different sources may have. Differently from Gangemi et al. [27], our approach aims at parsing specific type of textual data and, moreover, at combining information from various textual resources. For this purpose, we combined results of open domain information extraction tools with information related to the scholarly domain. Furthermore, within our approach more scientific papers are parsed in order to come up with knowledge resulting from the synthesis of various pieces of texts that refer to the same topic. With our approach the resulting scientific knowledge graph represents the overall knowledge presented within the input scientific publications.

Researchers have already targeted scientific publications as a challenge domain where to extract structured information [43, 44]. Furthermore, within the scholarly domain, extraction of relations from scientific papers has recently raised interest within the *SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications* [45] and SemEval 2018 Task 7 *Semantic Relation Extraction and Classification in Scientific Papers* challenge [46], where participants had to face the problem of detecting and classifying domain-specific semantic relations. Since then, extraction methodologies for the purpose to build knowledge graphs from scientific papers started to spread in literature [47, 48]. For example, Al-Zaidy et al. [49] employed syntactical patterns to detect entities, and defined two types of relations that may exist between two entities (i.e., *hyponymy* and *attributes*) by defining rules on noun phrases. Another attempt to build scientific knowledge graphs from scholarly data was performed by Yi and colleagues [29], as an evolution of authors' work at SemEval 2018 Task 7. First, authors proposed a Deep Learning approach to extract entities and relations from scientific literature. Then, they used the retrieved triples for building a knowledge graph on a dataset of 110,000 papers. Although our work takes inspiration from that, we propose different strategies to address open issues for combining entities and relations. For example, for solving ambiguity issues that regard the various representations of entities, Yi and colleagues [29] considered clusters of co-referenced entities to come up with a representative entity in the cluster. On the contrary, we adopted textual and statistics similarity to solve the ambiguity. Furthermore, they only used a set of predefined relations that might be too generic for the purpose of yielding insights from the research landscape. Within our approach we tried to detect relations that imply an action of an entity toward another one, making our results more precise and fine-grained.

[40] `http://wit.istc.cnr.it/stlab-tools/fred/`

6. Conclusions

In this paper we designed and developed a pipeline for representing the knowledge of scientific publication into a structured graph that we called scientific knowledge graph. We employed various state-of-the-art NLP tools and machine learning, and provided a workflow to merge their results. Moreover, we integrated the knowledge coming from many scientific publications into a single knowledge graph with the purpose to represent detailed knowledge of the scientific literature about the *Semantic Web* domain. The evaluation proved that this solution is able to automatically produce good quality scientific knowledge graphs and that the integration of different tools yields a better overall performance.

There are a number of limitations that need to be still addressed in future work. In the first instance, the current version does not take full advantage of the semantic characterization of the research entities to verify the resulting triples. For instance, it is currently possible for an entity of kind *Material* to include a entity of kind *Task*, which may be semantically incorrect. For this reason, we plan to develop a more robust semantic framework that could drive the extraction process and discard triples that do not follow specific constraints. For example, we could state that a material could include another material, but not a task or a method. These requirements could be enforced and verified with the use of specific semantic technologies for expressing constraints such as SHACL[41]. A second limitation is that the current prototype can only extract one relationship between two entities. This is not completely realistic since two entities can be linked by many kinds of relationships. This could also lead to a higher number of relationships that could suggest different applications or uses of entities, increasing the probability of finding unconsidered issues and solutions within a research field. We intend to explore this possibility in future work. Additionally, we will thoroughly investigate the conjunction construct which might hide rich knowledge about the relationship that frequently occurs between two research entities (e.g., *machine learning* and *data mining*). We also plan to improve the knowledge graph by considering cross document relations (e.g., citations) to further link our entities, in order to better support tools for scientific inquiry. A third limitation regards our ability to recognize synonyms that are not defined in existent knowledge bases, such as CSO. For instance, the current version may still fail to recognize that two quite different strings (e.g., Radial Basis Function Neural Network and RBFNN) actually refer to the same entity. We intend to address this issue by computing the semantic similarity between word and graph embeddings representing the entities in order to detect and merge synonyms more effectively. A fourth limitation regards the scalability of our pipeline. The current implementation presents a few bottlenecks that could make difficult to apply it on very large-scale datasets. First, the Extractor Framework requires a lot of hard disk space. This entails that data must be sampled to be processed. Second, the current pipeline only adopts the Stanford Core NLP server with just one thread, which requires a long time to mine textual resources sentence-by-sentence. However, this is not a big issue since it would be possible to run the Stanford Core NLP server in multi-thread

[41] https://www.w3.org/TR/shacl/

mode, speeding up the extraction process. An important next step will also be to perform an extrinsic evaluation of the proposed knowledge base within different tasks. In particular, we would like to assess how AI tasks such as those tackled by recommender systems or graph embeddings creation strategies can benefit from it.

Acknowledgements

Danilo Dessì acknowledges Sardinia Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2014-2020). This work has also been partially supported by a public grant overseen by the French National Research Agency (ANR) as part of the program "Investissements d'Avenir" (reference: ANR-10-LABX-0083). It contributes to the IdEx Université de Paris - ANR-18-IDEX-0001. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] M. Y. Jaradeh, S. Auer, M. Prinz, V. Kovtun, G. Kismihók, M. Stocker, Open research knowledge graph: Towards machine actionability in scholarly communication, arXiv preprint arXiv:1901.10816 (2019).

[2] D. Buscaldi, D. Dessì, E. Motta, F. Osborne, D. Reforgiato Recupero, Mining scholarly data for fine-grained knowledge graph construction, in: CEUR Workshop Proceedings, Vol. 2377, 2019, pp. 21–30.

[3] J. P. Tennant, H. Crane, T. Crick, J. Davila, A. Enkhbayar, J. Havemann, B. Kramer, R. Martin, P. Masuzzo, A. Nobes, et al., Ten hot topics around scholarly publishing, Publications 7 (2) (2019) 34.

[4] D. Buscaldi, D. Dessì, E. Motta, F. Osborne, D. R. Recupero, Mining scholarly publications for scientific knowledge graph construction, in: The Semantic Web: ESWC 2019 Satellite Events, 2019, pp. 8–12. doi:10.1007/978-3-030-32327-1_2.

[5] H. Kitano, Artificial intelligence to win the nobel prize and beyond: Creating the engine for scientific discovery, AI magazine 37 (1) (2016) 39–49.

[6] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs., SEMANTiCS (Posters, Demos, SuCCESS) 48 (2016).

[7] C. Lan, Y. Yang, X. Li, B. Luo, J. Huan, Learning social circles in ego networks based on multi-view social graphs, arXiv preprint arXiv:1607.04747 (2016).

[8] D. Dessì, J. Cirrone, D. R. Recupero, D. Shasha, Supernoder: a tool to discover over-represented modular structures in networks, BMC bioinformatics 19 (1) (2018) 318.

[9] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, et al., Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia, Semantic Web 6 (2) (2015) 167–195.

[10] A. G. Nuzzolese, A. L. Gentile, V. Presutti, A. Gangemi, Conference linked data: the scholarlydata project, in: ISWC, Springer, 2016, pp. 150–158.

[11] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, A. Kanakia, Microsoft academic graph: When experts are not enough, Quantitative Science Studies 1 (1) (2020) 396–413.

[12] Y. Zhang, F. Zhang, P. Yao, J. Tang, Name disambiguation in aminer: Clustering, maintenance, and human in the loop., in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1002–1011.

[13] P. Knoth, Z. Zdrahal, Core: three access levels to underpin open access, D-Lib Magazine 18 (11/12) (2012) 1–13.

[14] S. Peroni, D. Shotton, Opencitations, an infrastructure organization for open scholarship, Quantitative Science Studies 1 (1) (2020) 428–444.

[15] D. Shotton, Semantic publishing: the coming revolution in scientific journal publishing, Learned Publishing 22 (2) (2009) 85–94.

[16] A. G. Nuzzolese, A. L. Gentile, V. Presutti, A. Gangemi, Semantic web conference ontology-a refactoring solution, in: European Semantic Web Conference, Springer, 2016, pp. 84–87.

[17] F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, J. Morissette, Bio2rdf: towards a mashup to build bioinformatics knowledge systems, Journal of biomedical informatics 41 (5) (2008) 706–716.

[18] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, et al., The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud, Nucleic acids research 41 (W1) (2013) W557–W561.

[19] P. Groth, A. Gibson, J. Velterop, The anatomy of a nanopublication, Information Services & Use 30 (1-2) (2010) 51–56.

[20] T. Kuhn, C. Chichester, M. Krauthammer, N. Queralt-Rosinach, R. Verborgh, G. Giannakopoulos, A.-C. N. Ngomo, R. Viglianti, M. Dumontier, Decentralized provenance-aware publishing with nanopublications, PeerJ Computer Science 2 (2016) e78.

[21] J. Schneider, P. Ciccarese, T. Clark, R. D. Boyce, Using the micropublications ontology and the open annotation data model to represent evidence within a drug-drug interaction knowledge base, 2014.

[22] S. Peroni, D. Shotton, The spar ontologies, in: International Semantic Web Conference, Springer, 2018, pp. 119–136.

[23] S. Fathalla, S. Auer, C. Lange, Towards the semantic formalization of science, in: Proceedings of the 35th Annual ACM Symposium on Applied Computing, 2020, pp. 2057–2059.

[24] S. Peroni, F. Osborne, A. Di Iorio, A. G. Nuzzolese, F. Poggi, F. Vitali, E. Motta, Research articles in simplified html: a web-first format for html-based scholarly articles, PeerJ Computer Science 3 (2017) e132.

[25] S. Auer, V. Kovtun, M. Prinz, A. Kasprzik, M. Stocker, M. E. Vidal, Towards a knowledge graph for science, in: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, ACM, 2018, p. 1.

[26] S. Mesbah, C. Lofi, M. V. Torre, A. Bozzon, G.-J. Houben, Tse-ner: An iterative approach for long-tail entity extraction in scientific publications, in: ISWC, Springer, 2018, pp. 127–143.

[27] A. Gangemi, V. Presutti, D. R. Recupero, A. Nuzzolese, et al., Semantic Web Machine Reading with FRED, Semantic Web 8 (6) (2017) 873–893.

[28] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, A. B. Rios-Alvarado, Openie-based approach for knowledge graph construction from text, Expert Systems with Applications 113 (2018) 339–355.

[29] Y. Luan, L. He, M. Ostendorf, H. Hajishirzi, Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction, in: Proceedings of the EMNLP 2018 Conference, 2018, pp. 3219–3232.

[30] A. A. Salatino, F. Osborne, T. Thanapalasingam, E. Motta, The cso classifier: Ontology-driven detection of research topics in scholarly articles (2019).

[31] A. A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne, E. Motta, The computer science ontology: a large-scale taxonomy of research areas, in: ISWC, 2018, pp. 187–205.

[32] G. Angeli, M. J. J. Premkumar, C. D. Manning, Leveraging linguistic structure for open domain information extraction, in: Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP, Vol. 1, 2015, pp. 344–354.

[33] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[35] S. Angioni, F. Osborne, A. A. Salatino, D. Reforgiato, E. M. Recupero, Integrating knowledge graphs for comparing the scientific output of academia and industry, in: International Semantic Web Conference ISWC 2019, 2019, pp. 85–88.

[36] A. A. Salatino, F. Osborne, E. Motta, Augur: Forecasting the emergence of new research topics, in: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL '18, ACM, New York, NY, USA, 2018, pp. 303–312. doi:10.1145/3197026.3197052.

[37] A. Salatino, F. Osborne, E. Motta, Researchflow: Understanding the knowledge flow between academia and industry (2020).

[38] V. A. Carriero, A. Gangemi, M. L. Mancinelli, L. Marinucci, A. G. Nuzzolese, V. Presutti, C. Veninata, Arco ontology network and lod on italian cultural heritage., in: ODOCH@ CAiSE, 2019, pp. 97–102.

[39] G. Bardaro, D. Dessì, E. Motta, F. Osborne, D. R. Recupero, Parsing natural language sentences into robot actions., in: ISWC Satellites, 2019, pp. 93–96.

[40] S. Angioni, A. A. Salatino, F. Osborne, D. R. Recupero, E. Motta, Integrating knowledge graphs for analysing academia and industry dynamics, in: ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, Springer, 2020, pp. 219–225.

[41] A. Moro, A. Raganato, R. Navigli, Entity linking meets word sense disambiguation: a unified approach, Transactions of the Association for Computational Linguistics 2 (2014) 231–244.

[42] J. R. Curran, S. Clark, J. Bos, Linguistically motivated large-scale nlp with c&c and boxer, in: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, 2007, pp. 33–36.

[43] F. Ronzano, H. Saggion, Dr. inventor framework: Extracting structured information from scientific publications, in: N. Japkowicz, S. Matwin (Eds.), Discovery Science, Springer International Publishing, Cham, 2015, pp. 209–220.

[44] D. O'Donoghue, Y. Abgaz, D. Hurley, F. Ronzano, Stimulating and simulating creativity with dr inventor, in: Proceedings of the Sixth International Conference on Computational Creativity June 2015, Brigham Young University, Utah, 2015, pp. 220–227, this is the postprint version of the published chapter.
URL http://mural.maynoothuniversity.ie/6347/

[45] I. Augenstein, M. Das, S. Riedel, L. Vikraman, A. McCallum, SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 546–555. doi:10.18653/v1/S17-2091.
URL https://www.aclweb.org/anthology/S17-2091

[46] K. Gábor, D. Buscaldi, A.-K. Schumann, B. QasemiZadeh, H. Zargayouna, T. Charnois, Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers, in: Proceedings of The 12th International Workshop on Semantic Evaluation, 2018, pp. 679–688.

[47] A. Li, X. Wang, W. Wang, A. Zhang, B. Li, A survey of relation extraction of knowledge graphs, in: J. Song, X. Zhu (Eds.), Web and Big Data, Springer International Publishing, 2019, pp. 52–66.

[48] P. Labropoulou, D. Galanis, A. Lempesis, M. Greenwood, P. Knoth, R. E. de Castilho, S. Sachtouris, B. Georgantopoulos, S. Martziou, L. Anastasiou, K. Gkirtzou, N. Manola, S. Piperidis, Openminted: A platform facilitating text mining of scholarly content, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Paris, France, 2018.

[49] R. A. Al-Zaidy, C. L. Giles, Extracting semantic relations for scholarly knowledge base construction, in: 2018 IEEE 12th international conference on semantic computing (ICSC), IEEE, 2018, pp. 56–63.